COMPUTATIONAL LOGIC
A PROGRESS REPORT

J. A. Robinson

June, 1972

SYSTEMS AND INFORMATION SCIENCE
SYRACUSE UNIVERSITY

BEST
AVAILABLE COPY

Form Approved
Budget Bureau No. 22-R-293

Syracuse University Computational Logic Project

Semi Annual Technical Report No. 1

ARPA Order No: 1888
Contract No: DAHC04-72-C-0003
Program Code No: 61101D
Principal Investigator: J. A. Robinson
(315) 476-5541 Ext. 3159
Contractor: Syracuse University, Syracuse, NY 13210
Effective date of contract: August 25, 1972
Contract expiration date: August 24, 1973
Short title of work: Computational Logic Project
Amount of grant: $491,162.00

Sponsored by

Advanced Research Projects Agency

ARPA Order No. 1888

COMPUTATIONAL LOGIC:
A PROGRESS REPORT ON
THE SYRACUSE UNIVERSITY
COMPUTATIONAL LOGIC PROJECT

June, 1972

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1888

SUMMARY.

This report describes the research work that is in progress in the Computational Logic Project at Syracuse University. The period covered is from September 1, 1971 to May 31, 1972. A general introduction to the whole subject area is presented in the opening section, in which an attempt is made to view the field in a broad perspective and to relate it to the concerns of computer science and of artificial intelligence. The sections which follow the introduction are brief accounts of the particular problems which are currently under investigation. These problems are all related, in various ways, to the effort to design computational methods of inference-making for certain formal languages. In some cases the languages are previously developed standard systems such as the predicate calculi of first and higher orders; in other cases part of the problem is first to define a suitable formal language which will meet some given criteria, and then to elaborate a computational theory of inference for it upon which can be based efficient and useful algorithms.

In some of this work the results can be expected to lead to applications, but it should be emphasized that often the outcome is "merely" understanding the phenomena

involved, and discovering that the hard facts of the matter preclude any useful applications. One cannot in general predict which way a given investigation will pan out. The fundamental motivation is to find out the truth about how deductive reasoning works, in the hope that at least some aspects of it can be reproduced in computing machines.

COMPUTATIONAL LOGIC.

It has been one of the main goals of logicians, at least since the time of Leibniz, to reduce deduction problems to problems of calculation. Ideally, it was thought, one would eventually develop a universal algorithm for computing the answer to all questions of the form: "does A follow logically from B?". As late as 1930 this algorithm was thought to exist, and indeed its discovery was felt to be getting closer as a result of an intense burst of creative activity in mathematical logic in the early twentieth century.

In 1936 Alonzo Church proved that there is no such algorithm. We are therefore now engaged in a more modest quest, namely to try to find out to what extent deduction problems can be reduced to problems of computation.

The period 1930-1970 saw the development of a body of ideas leading to an algorithm of proof discovery for all those problems of form "does A follow logically from B?" which can be formulated within the first order predicate calculus (FOPC). That is, when A and B are sentences of FOPC, the algorithm is guaranteed to give the correct answer whenever the answer is yes, and indeed it provides as a bonus a piece of text which when properly interpreted is seen to be a deduction of B from A. From

about 1957 onwards a growing band of matehmaticians and computer scientists have been experimenting with variations of this algorithm, principally to see how efficient it can be made in practical engineering terms. Since about 1964 this work has been mainly devoted to attempts to exploit the so-called resolution method which was proposed in 1963 by J. A. Robinson as the most advantageous way of harnessing certain combinatorial possibilities which had been uncovered by Herbrand (1930) and analyzed more thoroughly by Kanger and Prawitz around 1960.

This resolution methodology is still under active investigation in many research centers, and a number of the problems described later in this report are to be understood as being part of this investigation. It appears that we are by no means yet at the end of exploring and applying the power inherent in the resolution method. Work at some centers, notably Stanford (Luckham et al.) indicates that increasingly more difficult deduction problems are being solved with the help of the method. One can at present see no obvious limitations of principle barring the way to still further improvement.

However, it has been for some time clear that far more needs to be done than merely to follow up resolution and related ideas, if the process of proof discovery is to be mechanized on a very much larger scale and with respect to deduction problems of wider scope and greater practical interest than those which have hitherto been

studied. It has become more and more obvious that logic is only a part of a much wider science - epistemology - which is concerned in general with knowledge: what it is, how it works, how it grows, how to formulate it and organise it formally, how to deploy it in machines. Accordingly the scope of logical research is seen to be far greater than the mathematical analysis of the relatively simple - although enormously powerful - formal languages like FOPC which have for over half a century served as models for the language of mathematical discourse. Actual mathematical discourse - let alone discourse about the physical world and human affairs - is immensely more complicated and sophisticated than can be adequately reflected in any known formal language. In order to study actual discourse it will be necessary to develop very much richer artificial language systems, systems much nearer in spirit and "feel" to the natural languages, but which are nevertheless formally and exactly defined. The best way to tackle this seems to be to begin with that which is to be modelled - actual discourse and knowledge systems - and to try to see what is there and how it works.

This is the motivation for the "case studies" work to be described later The idea in outline is to take some actual pieces of mathematical knowledge - in the form of

non-trivial theorems - and trace in detail their dependence upon other items of knowledge, all the way back to the simplest and most fundamental level. This network of dependence is not simply a "proof" of the theorem, although that is part of it. There is also - a much neglected aspect of exact epistemology - the derivation of the <u>concepts</u> involved from simpler ones. The tracing back of concepts which occur in a deep theorem is often much more important to an understanding of the theorem than the steps of what is usually taken to be its "proof".

Another way of making this same point is to remark that when a human mathematician demonstrates a proposition either to himself or for the benefit of others, what he is principally engaged in doing is setting up an appropriate context of knowledge, a conceptual environment within which the theorem becomes <u>immediately obvious</u>. This phenomenon, <u>obviousness and immediacy</u> of propositions from within suitable conceptual contexts, is of very great importance for an understanding of the process which is normally taken to be deductive reasoning.

It is as important to understand how a proof works as to understand how to discover it. This point has been little stressed in the traditional approach to logic, and equally neglected in modern attempts to devise efficient methods of <u>proof discovery</u> for the computer. Among our

longer-range goals we envisage the development of a computer program which will be capable of "following a mathematical narrative" in analogous fashion to the way in which one follows a story. The process involves the growing of a system of knowledge - the building-up of "cognitive structure" stage by stage as the proof unfolds. Many of the steps in this process correspond to definitions and suppositions rather than to conclusions. Often a proof is virtually completed by introducing the "right concept", such as the concept of alternate black and white coloring of the squares on the mutilated checkerboard. In the right framework of concepts, the proposition to be proved becomes obviously true, and requires no proof!

By very detailed examination of some actual cases it is hoped that some guidance can be obtained for the design of programs in which one can conduct transactions of this kind with a computer. We have to try to find out what it is about a step in reasoning which makes it possible for a human to follow it - what is "hard" and what is "easy", and why.

RESOLUTION METHODOLOGY.

Within the general framework of ideas associated with the term _resolution_ a number of research efforts have been initiated.

DIRECTED RESOLUTION.

Luis Sanchis has developed a principle of inference called _directed resolution_ which is currently being subjected to empirical test in the form of a computer program. The idea behind Sanchis' method is to specialise the resolution rule for each individual deduction problem, so that in seeking to deduce B from A each inference step considered will "aim at" B. This is in contrast to the basic principle itself, which is neutral with respect to all deduction problems and condones inferences which are "in the wrong direction". Sanchis' technique involves the imposition of restrictions on the resolvents which are permitted to be formed, which are so devised as to eliminate those which are not appropriate. These restrictions depend upon the content of A and B.

MAXIMAL MODULATION.

Two efforts have been concerned with the problem of extending the techniques and concepts of resolution to the first order predicate calculus, with equality (FOPCE), viz.,

the formal language obtained by adding to the usual predicate calculus of first order the equality symbol as a logical constant. E. E. Sibert has designed a system of inference rules for FOPCE which consists essentially of resolution with an added rule called maximal modulation which deals with the inferences that depend on the logical properties of the equality relation. Sibert has studied this system theoretically and is now planning to subject it to empirical test on the machine.

UNIFICATION AND EQUALITY.

The second of the two FOPCE efforts has been a detailed analysis of a result of R. E. Wengert which relates unification with equality logic in a novel manner. Wengert, Kowalski and Robinson have conducted an extended study of Wengert's result with a view to clarifying it and devising applications. The basic idea is this: it can be shown that, in FOPC, if we have an unsatisfiable set S of clauses then we can systematically find a set T of clauses, each of which is merely a variant of a clause in S, such that T is internally contradictory. In FOPC we say that a set T of clauses is internally contradictory if there is a substitution s such that Ts is a boolean unsatisfiable set. This property is a decidable one, and the importance of this basic result is that it leads directly to a proof procedure

for FOPC in which in a certain sense there is no redundancy in the search. In his 1970 Ph.D. thesis Wengert succeeded in extending this result to FOPCE. In doing so, he had to contrive suitable generalisations for the essential ideas involved, in particular he had to adapt to FOPCE the idea of being internally contradictory. In FOPCE this definition is changed only slightly; instead of boolean unsatisfiability one must refer to unsatisfiability depending upon both boolean and equality semantics. The big difference in the FOPCE case is that <u>some of the variants in the set T play no role in the unsatisfiability of the set Ts as such, but are present simply to serve as auxiliary material for the algorithm which computes the substitution s</u>. This seems to be a completely new phenomenon, and is not yet fully understood. It is felt that this work could lead to proof procedures for FOPCE which are of great interest and importance, and our study of it continues.

RESLAB.

Robinson and S. Davidson have written a package of LISP functions called RESLAB which is intended as a standard working programming library for the project's efforts in resolution programming. It is planned to issue this system as a fully documented package for use by anyone with access to a LISP system. The programs have been

designed to be as perspicuous and flexible as possible and they incorporate the most powerful versions known of the unification and subsumption algorithms. It is also planned to extend the RESLAB package to incorporate a suitable selection of modules which deal with equality logic.

HIGHER ORDER PROGRAMMING LANGUAGES.

J. C. Reynolds has been working on the theory of higher order programming languages, especially concentrating on the recent ideas of Dana Scott about lattices of continuous functions. Reynolds' work has in the past few months focussed on what he calls <u>definitional interpreters for higher order programming languages</u>. The following is the abstract of his forthcoming paper on this subject:

"Higher order programming languages (i.e., languages in which procedures or labels can occur as values) are usually defined by interpreters which are themselves written in a programming language based on the lambda-calculus (i.e., an applicative language such as pure LISP). Examples include McCarthy's definition of LISP, Landin's SECD machine, the Vienna definition of PL/I, Reynolds' definition of GEDANKEN, and recent unpublished work by L. Morris and C. Wadsworth. Such definitions

can be classified according to whether the interpreter contains higher-order functions, and whether the order of application (i.e., call by value versus call by name) in the defined language depends upon the order of application in the defining language. As an example, we consider the definition of a simple applicative programming language by means of an interpreter written in a similar language. Definitions in each of the above classification are derived from one another by informal but constructive methods. The treatment of imperative features such as jumps and assignment is also discussed."

SEMANTIC PARTITION METHOD FOR HIGHER ORDER PREDICATE CALCULUS.

In 1969 J. A. Robinson proposed (in Machine Intelligence 4) a method of mechanizing the semantics of the higher order predicate calculus (HOPC) which he called the semantic partition method. Actual implementation of this method is complex and difficult, and our current plans are to write a number of programs incorporating features of the semantic partition method before trying to organise the rather major program which will contain the whole system. Robinson and E. F. Storm have been active in this work.

It is in this work that we intend to get at the phenomenon mentioned in the introduction, namely, the incremental semantic acquisition of information in

narrative mode, as sentence after sentence is uttered in the course of a mathematical demonstration. The basic idea is that as each sentence is received, its semantic content is stored away in an internal structure which abstractly is a representation of the set of all those semantic partitions of the set of expressions so far encountered which are compatible with the restrictions contained in what has so far been asserted and/or stipulated. Relative to this internal structure, it can be rapidly determined, by evaluation within each partition, whether a new sentence immediately follows from all that has gone before. This is a model of what actually seems to happen in the case of human comprehension of a proof-narrative and his incremental assent to the claims that are made sequentially as the argument unfolds. In the envisaged implementation the machine will play the role of the listener and digester of the information. The objective is to model how mathematical demonstrations actually work, as opposed to how they are invented.

UNIFICATION IN HOPC.

Since the computational logic project started in September 1971, some important but unpublished results have been circulating informally concerning the difficult problem of designing a unification algorithm for the full HOPC which suitably generalises the well-known unification algorithm for FOPC. Tomasz Pietrzykowski

and Donald C. Jensen of the University of Waterloo have apparently succeeded in setting up such an algorithm. It is quite complicated and the associated theory is not entirely clear, but at Syracuse we are devoting some effort in our project to understanding it and adapting the ideas to our purposes. Robinson and J. Schwarz have been concerned with this. We propose to invite Pietrzykowski and Jensen to Syracuse for a visit later on in the summer, during which we will study the algorithm intensively and try to work out a program to test its efficiency. If the method works as advertised, it will probably prove to be a major step forward in the general effort to automate HOPC. Accordingly we place considerable weight on this particular investigation.

HILBERT'S EPSILON CALCULUS.

Robinson is currently spending a large part of his time on a plan to exploit an old idea of David Hilbert's which has never been fully appreciated and in particular has never been made the basis for a computational treatment of FOPC. Hilbert's idea is to eliminate quantifiers from FOPC with the help of a new primitive notion, namely, the formation, from any sentence S and variable x, of a <u>term</u>, (εxS). This term intuitively denotes an object of which the sentence S is true, if there be any such objects, and otherwise an arbitrary object in the universe of discourse. With this semantical convention, it is

possible to construe the usual quantifications:

$$(\exists xS), \quad (\forall xS)$$

as being nothing but syntactical abbreviations for the sentences

$$S\{(\varepsilon xS)/x\}, \quad S\{(\varepsilon x\neg S)/x\}$$

respectively. Any sentence of FOPC can thus be translated automatically into one in which there are no quantifiers, but which consists of only atomic sentences built from only proper names and constant terms, combined into sentences by equality symbols and the usual boolean connectives. For sentences of this latter kind one can devise attractively simple "ground" algorithms for testing for satisfiability, especially with the framework of the semantic partition point of view already mentioned. However, there are some very nasty problems associated with such an enterprise. The translation into "pure epsilon calculus" of a sentence involving nested quantification will, if done in a straightforward manner, produce an equivalent quantifier-free sentence which is of enormous length. One pays for the elegance of the quantifier-free language by incredible complexity of the compound names which are needed in the typical assertion. So the problem is to get around this by choosing a better representation for the translated sentences than Hilbert's original one - which

in fact has always been a purely theoretical notation, not intended for serious practical use. It is felt that the basic idea of Hilbert's epsilon calculus is of very great importance and that it has tremendous potential for the mechanization of deductive reasoning. The "epsilon terms" correspond exactly to the intuitive phrases which enter naturally into mathematical discourse when a "typical entity" is introduced to play the part, in the subsequent argument, of all the entities having a given property.

CASE STUDIES IN MATHEMATICAL REASONING.

A start has been made by Robinson and Gordon D. Plotkin on an investigation of some actual cases, taken from the mathematical literature, of extended exposition of non-trivial mathematical material. The case that they have taken as the initial one for study is the exposition, by Hardy and Wright and also by LeVeque, of a proof by Erdos of a proposition in number theory known as "Bertrand's Conjecture". This proposition states that one can always find a prime number strictly between n and 2n, for every positive integer n greater than 1. Bertrand in 1845 verified the proposition by computation for all n not exceeding 3,000,000, and Chebyshev proved it for all n in 1850. The reason this case was chosen is that it is a very simple proposition to state and understand, yet is apparently

a rather difficult one to prove and certainly is hard to believe (it seems too good to be true!). It turns out that there are many concepts involved in the proof, which have to be understood in order to follow the reasoning, which are not needed for weaker results of the same kind, e.g., for the classical proposition of Euclid that one can always find a prime number between n and n! + 1. When Plotkin arrives in August for a postdoctoral stay on the project, he and Robinson will push this investigation forward and probably begin others of the same character.

THEORY OF HEURISTIC SEARCH.

Sibert and Donald Michie have completed an investigation into a problem arising in abstract heuristic search theory from the contemplation of the sort of search spaces one has to deal with in theorem proving problems. The following is the abstract of their forthcoming paper on the subject:

"Previous studies of heuristic search techniques have usually considered situations for which the search space could be represented as a tree, which limits the applicability of the results obtained. Here we offer a more general formulation in terms of derivation graphs, which correspond rather naturally to the search problems arising in automatic theorem proving and other areas. We consider a family of

search procedures controlled by evaluation functions of a very general sort, having the form $f_L(x, L_k)$, where L_k is that portion of the graph generated thus far by the procedure, and the node x is a candidate for incorporation into L_k. Completeness and minimality results are obtained for a number of procedures in this family, including methods analogous to those of Moore, Dijkstra, and Pohl."

PLANNER-LIKE SYSTEMS.

Jerry Schwarz has been investigating the ideas immanent in PLANNER and similar systems, and has launched an effort which falls into that category. He has under developement a system which attempts to combine the idea of state-space search with those of more general search-programming systems such as PLANNER. His system contains two major structures - a "smart" data base, and a plan. Both are specified using a directed graph called the planning graph. Nodes of the graph represent situations which will be encountered during the execution of the plan. Arcs represent transformations. Starting with a graph containing two nodes the system expands the graph by user-defined operators until a satisfactory plan is generated. The operators can use the "smart" data base which contains statements of a language for set relations, and which can also make inferences about those relations. The system should be applicable to such tasks as "robot planning" (a la STRIPS), automatic program writing, and simple game playing.

STRIPS-LIKE SYSTEMS.

Michie and Robinson conducted an investigation into the Stanford Research Institute's STRIPS problem solving system, with a view to developing an improved system of the same character. Namely, a "world" is given in which a robot device must operate in such a way as to achieve some given objective. The laws governing events and objects in the world are supplied in the form of sentences in some formal system, and the program computes a plan for the robot to carry out so as to reach the given goal. It emerged from the investigation that it might be possible to enrich considerably the concept of a plan, in the direction of a fully specified program written in, say, a PLANNER-like programming language. Robot-planning problems, after all, are simply problems calling for the automatic writing of programs within suitable programming languages. However, in the execution of such programs, it might be necessary to take into account explicitly the robot's state of knowledge as well as the state of the external world, and to classify some of the robot's operations as <u>epistemic</u>, i.e., aimed at changing the robot's state of knowledge, as opposed to <u>causal</u>, i.e., aimed at changing the state of the external world only. This work is at present incomplete, but progress to date was reported on at the 7th Machine Intelligence Workshop in June 1972.